# Lecture 1b: Markov Decision Processes
# CS885 Reinforcement Learning

2025-01-07

Complementary readings: [SutBar] Chap. 3, [Sze] Chap. 2, [RusNor] Sec. 15.1, 17.1-17.2, 17.4, [Put] Chap. 2, 4, 5

Pascal Poupart
David R. Cheriton School of Computer Science

UNIVERSITY OF WATERLOO

# Outline

- Markov Decision Processes
- Value Iteration

# Markov Decision Process

| Components | Formal definition | Inventory management |
|---|---|---|
| States | $s \in S$ | inventory levels |
| Actions | $a \in A$ | {doNothing, orderWidgets} |
| Rewards | $r \in \mathbb{R}$ | Profit ($) |
| Transition model | $\Pr(s_t \| s_{t-1}, a_{t-1})$ | Stochastic demand |
| Reward model | $\Pr(r_t \| s_t, a_t)$<br>$R(s_t, a_t) = \sum_{r_t} r_t \Pr(r_t \| s_t, a_t)$ | $R(s_t, a_t) =$ sales – costs – storage |
| Discount factor | $0 \le \gamma \le 1$ | $\gamma = 0.999$ |
| Horizon | $h \in \mathbb{N}$ or $h = \infty$ | $h = \infty$ |

# Common Assumptions

- Transition model
  - **Markovian**: $\Pr(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \ldots) = \Pr(s_{t+1}|s_t, a_t)$
    - Current inventory and order sufficient to predict future inventory
  - **Stationary**: $\Pr(s_{t+1}|s_t, a_t)$ is same for all $t$
    - Distribution of demand same every day
- Reward model
  - **Stationary**: $R(s_t, a_t) = \sum_t r_t \Pr(r_t|s_t, a_t)$ is same for all $t$
    - Formula to compute profits is same every day
  - **Exception:** terminal reward is often different
    - In a game: 0 reward at each step and
      +1/-1 reward at the end for winning/losing



UNIVERSITY OF WATERLOO

# Discounted/Average Rewards

- Goal: maximize total rewards $\sum_{t=0}^{h} R(s_t, a_t)$
  Problem: if $h = \infty$, then $\sum_{t=0}^{h} R(s_t, a_t)$ may be infinite

- Solution 1: discounted rewards
  - Discount factor: $0 \leq \gamma < 1$
  - Finite utility: $\sum_t \gamma^t R(s_t, a_t)$ is a geometric sum
  - $\gamma$ induces an inflation rate of $1/\gamma - 1$ (prefer utility sooner than later)

- Solution 2: average rewards
  - More complicated computationally (beyond scope of this course)

# Policy

- Choice of action at each time step
- Formally:
  - Mapping from states to actions: $\pi(s_t) = a_t$
  - Assumption: fully observable states
    - Allows $a_t$ to be chosen only based on current state $s_t$

# Policy Optimization

- **Policy evaluation:** compute expected utility

$$V^{\pi}(s_0) = \sum_{t=0}^{h} \gamma^t \sum_{s_{t+1}} \Pr(s_{t+1}|s_0, \pi)\, R(s_{t+1}, \pi(s_{t+1}))$$

- **Optimal policy $\boldsymbol{\pi}^*$:** policy with highest expected utility

$$V^{\pi^*}(s_0) \geq V^{\pi}(s_0) \;\; \forall \pi$$

- Several classes of algorithms:
  - Value iteration
  - Policy iteration
  - Linear Programming
  - Search techniques

# Value Iteration

- Value when no time left:

$$V_0^*(s_h) = \max_{a_h} R(s_h, a_h)$$

- Value with one time step left:

$$V_1^*(s_{h-1}) = \max_{a_{h-1}} R(s_{h-1}, a_{h-1}) + \gamma \sum_{s_h} \Pr(s_h | s_{h-1}, a_{h-1}) \, V_0^*(s_h)$$

- Value with two time steps left:

$$V_2^*(s_{h-2}) = \max_{a_{h-2}} R(s_{h-2}, a_{h-2}) + \gamma \sum_{s_{h-1}} \Pr(s_{h-1} | s_{h-2}, a_{h-2}) \, V_1^*(s_{h-1})$$

- …
- Bellman's equation:

$$V_\infty^*(s_t) = \max_{a_t} R(s_t, a_t) + \gamma \sum_{s_{t+1}} \Pr(s_{t+1} | s_t, a_t) \, V_\infty^*(s_{t+1})$$

$$a_t^* = \operatorname*{argmax}_{a_t} R(s_t, a_t) + \gamma \sum_{s_{t+1}} \Pr(s_{t+1} | s_t, a_t) \, V_\infty^*(s_{t+1})$$

# Value Iteration

**valueIteration(MDP)**

$V_0^*(s) \leftarrow \max_a R(s,a) \;\; \forall s$

For $n = 1$ to $h$ do

$V_n^*(s) \leftarrow \max_a R(s,a) + \gamma \sum_{s'} \Pr(s'|s,a)\, V_{n-1}^*(s') \;\; \forall s$

Return $V^*$

Optimal policy $\pi^*$

$n = 0$: $\pi_0^*(s) \leftarrow \operatorname{argmax}_a R(s,a) \; \forall s$

$n > 0$: $\pi_n^*(s) \leftarrow \operatorname{argmax}_a R(s,a) + \gamma \sum_{s'} \Pr(s'|s,a)\, V_{n-1}^*(s') \;\; \forall s$

NB: $\pi^*$ is non-stationary (i.e., time dependent)

# Value Iteration (Matrix Form)

$R^a$: $|S|\times 1$ column vector of rewards for $a$

$V_n^*$: $|S|\times 1$ column vector of state values

$T^a$: $|S|\times|S|$ matrix of transition probabilities for $a$

**valueIteration(MDP)**

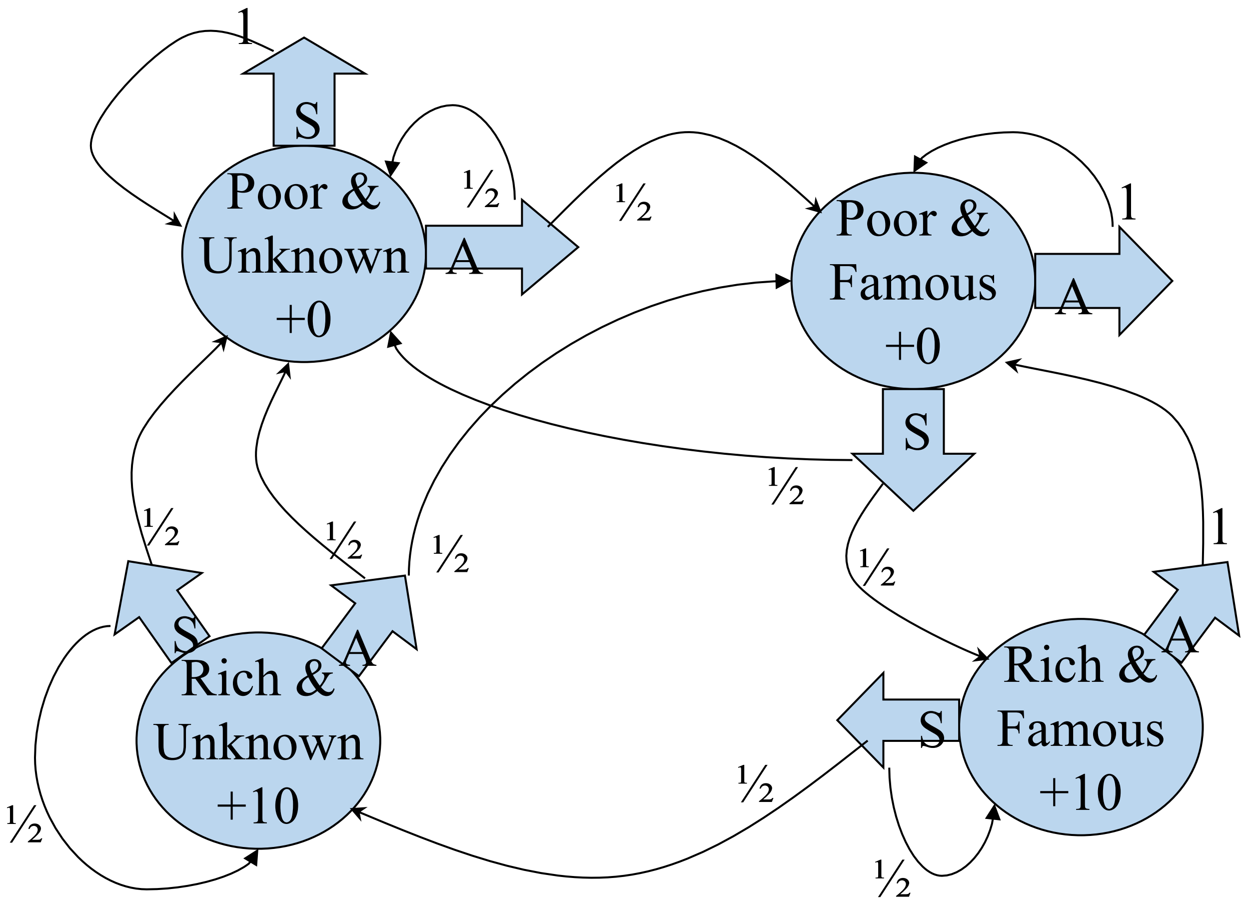
$V_0^* \leftarrow \max_a R^a$

For $t = 1$ to $h$ do

$$V_n^* \leftarrow \max_a R^a + \gamma T^a V_{n-1}^*$$

Return $V^*$

# A Markov Decision Process

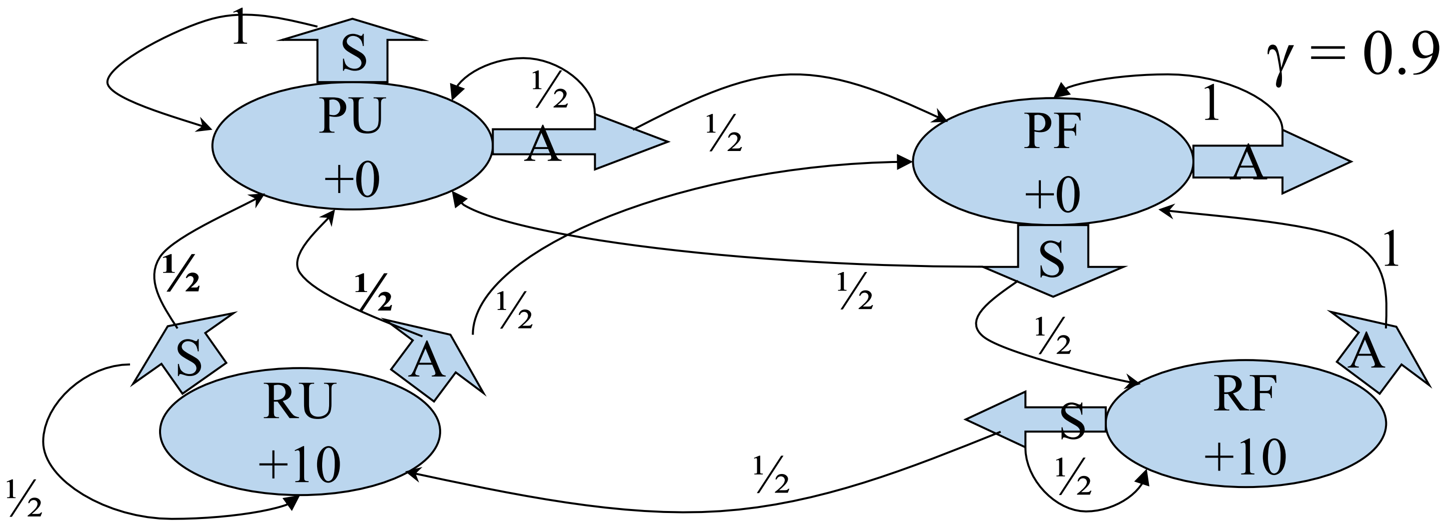



$\gamma = 0.9$

You own a company

In every state you must choose between

**S**aving money or **A**dvertising




| $n$ | $V^*_n(PU)$ | $\pi^*_n(PU)$ | $V^*_n(PF)$ | $\pi^*_n(PF)$ | $V^*_n(RU)$ | $\pi^*_n(RU)$ | $V^*_n(RF)$ | $\pi^*_n(RF)$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | A,S | 0 | A,S | 10 | A,S | 10 | A,S |
| 1 | 0 | A,S | 4.5 | S | 14.5 | S | 19 | S |
| 2 | 2.03 | A | 8.55 | S | 16.53 | S | 25.08 | S |
| 3 | 4.76 | A | 12.20 | S | 18.35 | S | 28.72 | S |
| 4 | 7.63 | A | 15.07 | S | 20.40 | S | 31.18 | S |
| 5 | 10.21 | A | 17.46 | S | 22.61 | S | 33.21 | S |

# Exercise: Value Iteration, No Time Left (RF State)

# Exercise: Value Iteration, One Time Step Left (RF State)

# Horizon Effect

- Finite $h$:
  - Non-stationary optimal policy
  - Best action different at each time step
  - Intuition: best action varies with the amount of time left
- Infinite $h$:
  - Stationary optimal policy
  - Same best action at each time step
  - Intuition: same (infinite) amount of time left at each time step
  - **Problem**: value iteration does infinite # of iterations



UNIVERSITY OF WATERLOO

# Infinite Horizon

- Assuming a discount factor $\gamma$, after $n$ time steps, rewards are scaled down by $\gamma^n$
- For large enough $n$, rewards become insignificant since $\gamma^n \to 0$

- Solution #1:
  - pick large enough $n$ and run value iteration for $n$ steps
  - Execute policy $\pi_n$ found at the $n^{th}$ iteration

- Solution #2:
  - Continue iterating until $||V_n - V_{n-1}||_\infty \leq \epsilon$ ($\epsilon$ is called tolerance)
  - Execute policy $\pi_n$ found at the $n^{th}$ iteration



UNIVERSITY OF WATERLOO